

On-line Handwritten Devanagari Character Recognition using Fuzzy Directional Features

Sunil Kumar Kopparapu, Lajish VL
SunilKumar.Kopparapu@TCS.Com, Lajish.VL@TCS.Com

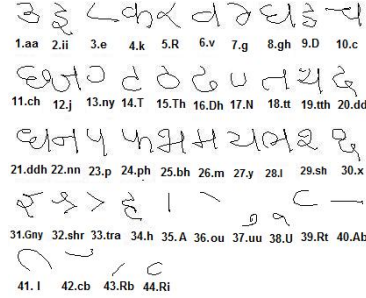
TCS Innovation lab - Mumbai,
Yantra Park, Tata Consultancy Services, Thane (West), Maharastra, India

Abstract. This paper describes a new feature set for use in the recognition of on-line handwritten Devanagari script based on Fuzzy Directional Features. Experiments are conducted for the automatic recognition of isolated handwritten character primitives (sub-character units). Initially we describe the proposed feature set, called the Fuzzy Directional Features (FDF) and then show how these features can be effectively utilized for writer independent character recognition. Experimental results show that FDF set perform well for writer independent data set at stroke level recognition. The main contribution of this paper is the introduction of a novel feature set and establish experimentally its ability in recognition of handwritten Devanagari script.

1 Introduction

Interest in on-line handwritten script recognition has been active for a long time. In the case of Indian languages, research work is active especially for Devanagari [6, 9], Bangala [10, 3], Telugu [1] and Tamil [12, 2] to name a few. In English script, the mostly widely researched, barring a few alphabets, all the alphabets can be written in a single stroke. But most of the Indian languages have characters which are made up of two or more strokes which makes it necessary to analyze a set of strokes to identify the entire character. We identified, through visual inspection of the script, a basis like set of 44 strokes¹ called *primitives* which are sufficient to represent all the characters in Devanagari script. The set of primitives used to write the complete Devanagari character set are shown in Figure 1(a). In this paper we use these primitives as the units for recognition. In an unconstrained handwriting these primitive strokes exhibit large variability in shape, direction and order of writing. A sample set of primitives collected from different writers is shown in Figure 1(b) to capture the variability in the way primitives are written. The variations within the primitives even for the same writer is evident and it is observed that the variation among different writers is even larger; making the task of recognizing these primitives difficult.

¹ Usually the segment of pen motion from the pen-down to the pen-up position is a loose definition of a stroke



(a) Primitive that can be used to write the complete alphabet set in Devanagari.

Sr. No.	Character	Handwritten Samples					
		1	2	3	4	5	6
1.	ॐ	ॐ	ॐ	ॐ	ॐ	ॐ	ॐ
2.	ई	ई	ई	ई	ई	ई	ई
3.	८	८	८	८	८	८	८
4.	ॡ	ॡ	ॡ	ॡ	ॡ	ॡ	ॡ
5.	ॢ	ॢ	ॢ	ॢ	ॢ	ॢ	ॢ
6.	ॣ	ॣ	ॣ	ॣ	ॣ	ॣ	ॣ
7.	।	।	।	।	।	।	।
8.	॥	॥	॥	॥	॥	॥	॥
9.	०	०	०	०	०	०	०
10.	१	१	१	१	१	१	१

(b) Variability in writing primitives.

Fig. 1. (a) Primitive handwritten strokes, (b) Wide variability is observed

The main challenge in on-line handwritten character recognition in Indian language is the large size of the character set, variation in writing style (when the same stroke is written by different writers or the same writer at different times) and the similarity between different characters in the script.

In this paper, we propose the use of Fuzzy Directional Features (FDF) set for the recognition of the primitives (which are also strokes). The variations that exist in the primitives (see Figure 1(b)) test the ability of the proposed features to recognize handwritten script. The rest of the paper is organized as follows. We introduce the Fuzzy Directional Features set in Section 2. Experimental results are outlined in Section 3, and conclusions are drawn in Section 4.

2 Fuzzy Directional Features Extraction

Several temporal features have been used for script recognition in general[7, 5, 11, 4] and for on-line Devanagari script recognition in particular. We propose a simple yet effective feature set based on fuzzy directional feature set². The detailed procedure for obtaining these directional features is given below.

Let an on-line handwritten character be represented by a variable number of 2D points which are in a time sequence. For example an on-line script would be represented as $\{(x_{t_1}, y_{t_1}), (x_{t_2}, y_{t_2}), \dots, (x_{t_n}, y_{t_n})\}$ where, t denotes the time and $t_1 < t_2 < \dots < t_n$. Equivalently we can represent the on-line character (see Fig 2(a), 2(c)) as

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

by dropping the variable t . The number of points denoted by n vary depending on the size of the character and also the speed with which the character is

² Note that [8] talks of fuzzy feature set for Devanagari script albeit for offline script

written. Most script digitizing devices (popularly called electronic pen) sample the script uniformly in time, generally at 100 Hz. For this reason, the number of sampling points is large when the writing speed is slow which is especially true at curvatures (see Figure 2(a), 2(c)); we exploit these curvature points in extracting FDF.

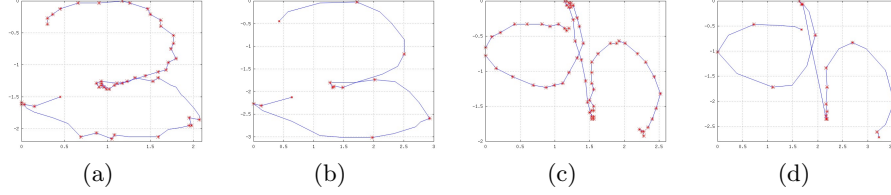


Fig. 2. Sample character (a, c) and its curvature points (b, d).

We first identify the curvature points (called critical points) from the smoothed (we use discrete wavelet transform) handwriting data. The sequence $(x_i, y_i)_{i=0}^n$ represents the handwriting data of a stroke. We treat the sequence x_i and y_i separately and calculate the critical points for each of these time sequence. For the x sequence, we calculate the first difference $x'_i = \text{sgn}(x_i - x_{i+1})$ where $\text{sgn}(\beta) = +1$ if $\beta > 0$, $\text{sgn}(\beta) = -1$ if $\beta < 0$ and $\text{sgn}(\beta) = 0$ if $\beta = 0$. We use x' to compute the critical point. The point i is a critical point if $x'_i - x'_{i+1} \neq 0$. Similarly we calculate the critical points for the y sequence. The final list of critical points is the union of all the points marked as critical points by both the x and the y sequence (see Figure 2(b), 2(d)). It must be noted that the position and number of curvature points computed for different samples of the same strokes vary. Trimming of curvature points is carried out on the obtained $k - 1$ direction sequence by removing all *spurious* curvature point. A curvature point is said to be spurious if a set of three curvature points results in the same direction. For the sake of discussion lets assume that there are no spurious curvature points.

Let k be the number of curvature points (denoted by c_1, c_2, \dots, c_k) extracted from a stroke of length n ; usually $k \ll n$. The k critical points form the basis for extraction of the FDF. We first compute the angle between the two curvature points, say c_l and c_m , as

$$\theta_{lm} = \tan^{-1} \left(\frac{y_l - y_m}{x_l - x_m} \right)$$

where (x_l, y_l) and (x_m, y_m) are the coordinates corresponding to the curvature point c_l and c_m respectively. The FDF set is computed using θ_{lm} . We use Algorithm 1 (assisted by triangular membership function, Algorithm 2) to compute the FDF. Note that every θ_{lm} (represented by θ in Figure 3(a) is the angle the blue dotted line makes with the 0° axis) has two directions (say $d_{lm}^1 = 1$, $d_{lm}^2 = 2$, note that the line in dotted blue in Figure 3(a) lies in both the triangles represented by direction 1 and direction 2) associated with it having m_{lm}^1, m_{lm}^2

membership values respectively (represented by the green and the red dot respectively in Figure 3(a)). Also note (a) $m_{lm}^1 + m_{lm}^2 = 1$ and (b) d_{lm}^1, d_{lm}^2 are adjacent directions, for example if $d_{lm}^1 = 5$ then d_{lm}^2 could be either 4 or 6.

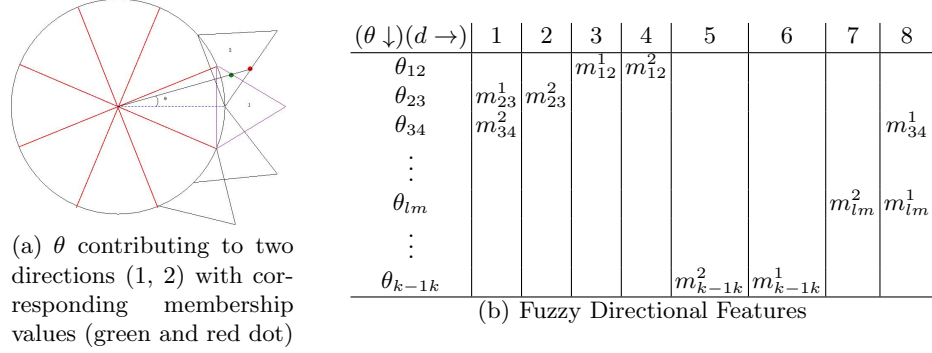


Fig. 3. Fuzzy Directional Features

Algorithm 1 Computing Fuzzy Directional Features

```

deg2fuzzydir(double  $\theta$ )
i=1; d[i] = -1; m[i] = -1
if ( $\theta > -\pi/4$  &  $\theta < \pi/4$ ) then
    d[i] = 1; m[i] = fuzzy_membership(0, $\theta$ );
end if
if ( $\theta \geq 0$  &  $\theta < 2\pi/4$ ) then
    d[i] = 2; m[i] = fuzzy_membership( $2\pi/4$ , $\theta$ );
    i++;
end if
 $\vdots$ 
{Similarly for  $d[i] = 3, 4, 5, 6, 7$ }
if ( $\theta > -2\pi/4$  &  $\theta < 0$ ) then
    d[i] = 8; m[i] = fuzzy_membership(0, $\theta$ );
    i++;
end if
return(d[i], m[i]);

```

It should be noted that the sum of the membership functions of a particular row (see Figure 3(b)) is always 1. Given an on-line character, we extract the FDF shown in Figure 3(b). Then we calculate the mean FDF by averaging across the columns, so as to form a vector of dimension 8. The mean is calculated as follows; for each direction (1 to 8), collect all the membership values and divide by the number of occurrences of the membership values in that direction. For example,

Algorithm 2 Triangular Fuzzy Membership Function

```
fuzzy_membership( $\theta_c, \theta$ )  
 $m = 1.0 - \frac{(|(\theta_c - \theta)|)}{(\pi/4)}$ ;  
return(m)
```

for Figure 3(b), the mean for direction 1 is calculated as $f_1 = \frac{(m_{23}^1 + m_{34}^1)}{2}$. In all our experiments we have used this mean FDF

$$\mathcal{F} = [f_1, f_2, \dots, f_8] \quad (1)$$

to represent a stroke.

3 Experimental Analysis

For experimental analysis, we collected handwritten data from 10 persons, each of whom wrote all the primitives of Devanagari text using Mobile e-Notes Taker³. This raw stroke data is smoothed using Discrete Wavelet Transform (DWT) decomposition⁴ to remove noise in terms of small undulation due to the sensitive-ness of the sensors on the electronic pen. For each stroke we extracted the fuzzy directional feature set as described in Section 2. We used 5 user data for training and the other 5 for the purpose of testing the performance of the FDF set. We initially hand tagged each stroke in the collected data using the 44 primitives that we selected (see Figure 1(a)).

For training, we calculated (1) for all strokes corresponding to the same primitive and computed the average to model the primitive. So a primitive was represented by a vector of size 8 by taking the average over all the occurrences of the primitive in the training set. All the experimental results are based on this data set (from 10 different writers).

For testing purpose, we took a stroke (t) to be recognized, we first extracted FDF (using Algorithm 1) and computed the mean FDF using (1). Then we compared it with the FDF model of the 44 reference strokes using the usual Euclidean distance measure. We computed for the test stroke t , its distance from all the primitives, namely, $\|F_t - F_i\|^2$ for $i = 1, \dots, 44$ and arranged them in the increasing order of magnitude (best match first). The results for this are shown in Table 1 for both the train data and the test data for $\alpha = 1, 2, 5$. Note that the values in Table 1 are computed as follow. For $N = \alpha$, the test stroke t is recognized as the primitive l if l occurs atleast at the α^{th} position from the best match (this is generally called the N-best in literature). As expected the recognition accuracies are poor (very similar to the phoneme recognition by a speech engine) for $\alpha = 1$ and improves with increasing α . It should be noted that the accuracies are writer independent and for stroke level recognition.

³ http://www.hitech-in.com/mobile_e-note_taker.htm

⁴ We do not dwell on this since this is well covered in pattern recognition literature.

Table 1. Recognition accuracies for train and test data set.

Data	$\alpha = 1$	$\alpha = 2$	$\alpha = 5$
Train Data	63.0% (139/220)	87.9% (193/220)	93.3%(205/220)
Test Data	37.0% (82/220)	54.6% (120/220)	78.1% (172/220)

4 Conclusions

In this paper we have introduces a new on-line script feature set, called the Fuzzy Directional Features. We have evaluated the performance of the novel feature set by presented recognition accuracies for writer independent stroke level data set. It is well known, both in speech and script recognition literature that stroke (phoneme in case of speech) recognition is always poor. As in speech we plan use (a) Viterbi traceback to enhance alphabet (multiple stroke) recognition and/or (b) cluster strokes using spatio-temporal information to form alphabets and then use the cluster of strokes to recognize them. This we believe will lead to better accuracies of writer independent script recognition.

References

1. Babu, V., Prasanth, L., Sharma, R., Rao, G., Bharath, A.: HMM-based online handwriting recognition system for Telugu symbols. In: ICDAR07. pp. 63–67 (2007)
2. Bharath, A., Madhvanath, S.: Hidden markov models for online handwritten Tamil word recognition. In: ICDAR07. pp. 506–510 (2007)
3. Bhattacharya, U., Gupta, B., Parui, S.: Direction code based features for recognition of online handwritten characters of Bangla. In: ICDAR07. pp. 58–62 (2007)
4. Connell, S., Jain, A.: Template-based online character recognition. Pattern Recognition 34(1), 1–14 (January 2001)
5. Garcia Salicetti, S., Dorizzi, B., Gallinari, P., Wimmer, Z.: Maximum mutual information training for an online neural predictive handwritten word recognition system. IJDAR 4(1), 56–68 (2001)
6. Joshi, N., Sita, G., Ramakrishnan, A., Deepu, V., Madhvanath, S.: Machine recognition of online handwritten Devanagari characters. In: ICDAR05. pp. II: 1156–1160 (2005)
7. Menier, G., Lorette, G., Gentric, P.: A genetic algorithm for on-line cursive handwriting recognition. In: ICPR. pp. B:522–525 (1994)
8. Mukherji, P., Rege, P.P.: Fuzzy stroke analysis of devnagari handwritten characters. W. Trans. on Comp. 7(5), 351–362 (2008)
9. Namboodiri, A., Jain, A.: Online handwritten script recognition. PAMI 26(1), 124–130 (January 2004)
10. Parui, S., Guin, K., Bhattacharya, U., Chaudhuri, B.: Online handwritten Bangla character recognition using HMM. In: ICPR08. pp. 1–4 (2008)
11. Schenk, J., Rigoll, G.: Neural net vector quantizers for discrete hmm-based on-line handwritten whiteboard-note recognition. In: ICPR. pp. 1–4 (2008)
12. Sundaram, S., Ramakrishnan, A.: A novel hierarchical classification scheme for online Tamil character recognition. In: ICDAR07. pp. 1218–1222 (2007)